The SIP protocol suite

How one of the most popular protocol suites for VoIP works
Introduction

- Session Initiation Protocol
  - Proposed in IETF
- Brand new protocol, not an adaptation of existing ones
  - Designed to take full advantage of IP features
- Application layer protocol
Features(1)

- “Limited” scope
  - Control protocol (signaling)
    - It does not specify how audio/video should be encoded
    - It does not make resource reservation
      - But it provides the user with the information needed to do it
  - Simpler and lighter
  - End-to-end signaling
    - SIP packets are not distinguished by routers from other IP datagrams
  - It uses other protocols, when available
    - SAP, SDP, RSVP, RTP/RTCP, ...

- Support for
  - Name mapping (ail address, etc)
  - Personal mobility
    - Calling a phone set, voice mail, e-mail, ...
Features (2)

- Client – server interaction
- HTTP-like message format (plain text messages)
- TCP, TLS e UDP
  - UDP: simple (e.g. device with limited memory and CPU power)
  - TCP: useful, for example, to cross firewalls
  - Reliability of the communication also over UDP
    - Three way handshake (Request, Response, Ack)
  - TCP: the same connection may be used for several request/response interactions
  - TLS: secure, but the advantage of text messages is lost

- No fragmentation
  - One message in a single packet
  - Size = MTU of the network, often 1500 bytes
  - Some critical aspects connected with the trand to introduces new headers, and to use existing protocols in a context they were not design to work with
Services supported in SIP

- **Voice calls**
  - Multi-party calls to be improved

- **E-presence**

- **Instant Messaging**

- **Whiteboard**
  - Few implementations

- **File transfer, interactive games**
  - Initial implementations, to be improved

- **Implementations biased towards VoIP, economical interest are bigger**
  - This choice may make difficult to provide value-added services
Main services for voice calls in VoIP

- **User localization**
  - Defines the (destination) terminal that should be used for the call

- **User capacity**
  - Defines the media (audio, video, ...) and the parameters (codec) to use

- **User availability**
  - Defines whether or not the other party is willing to communicate

- **Call setup**
  - Establishes a connection with all its parameters

- **Call management**
  - Additional services
  - Call management and transfer
The SIP protocol stack
SDP (Session Description Protocol)

- **Used to describe multimedia sessions**
  - Number of multimedia streams
  - Media type (audio, video, ...)
  - Codec (H.261 video, MPEG video, ...)
  - Transport protocol (RTP/UDP/IP, H.320, ...)
  - Bandwidth
  - Addresses and ports
  - Start/end times of each stream
    - Useful in case of media streaming or invitation for an audio/video conference
  - Source identification

- **Some non-useful/unused/obsolete parameters**
  - SDP adopted without modification to re-use existing software

- **Text format**

- **Included in the body of SIP messages**
SDP message format

- Divided in several sections
  - Session
    - A line starting with “v = ”
    - The parameters defined in this section are applicable for all the media, unless explicitly re-defined later
    - It includes at least one line devoted to “time information”
  - Media
    - A line starting with “m = ”
    - It is optional; it may be not used, or it appears N times (N equal to the number of media used)

```
Session
  v = 0
  ...

Media
  m = audio 3456 RTP/AVP 96
  ...
```
### SDP attributes

#### Session description fields

<table>
<thead>
<tr>
<th>V</th>
<th>SDP version</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>Session creator or session identifier</td>
</tr>
<tr>
<td>S</td>
<td>Session name</td>
</tr>
<tr>
<td>*I</td>
<td>Session information</td>
</tr>
<tr>
<td>*U</td>
<td>URI with the session description</td>
</tr>
<tr>
<td>*E</td>
<td>E-mail address</td>
</tr>
<tr>
<td>*P</td>
<td>Telephone number</td>
</tr>
<tr>
<td>*C</td>
<td>Connection information</td>
</tr>
<tr>
<td>*B</td>
<td>Session bandwidth in Kbps; it can be of CT type (Conference Total), or AS type (Application-Specific Maximum)</td>
</tr>
<tr>
<td>*X</td>
<td>Local time</td>
</tr>
<tr>
<td>*K</td>
<td>Encryption key; it applies to the whole session or to a specific media (auto, video)</td>
</tr>
<tr>
<td>*A</td>
<td>Zero or more lines with session attributes</td>
</tr>
</tbody>
</table>

#### Timing description

<table>
<thead>
<tr>
<th>R</th>
<th>Number of sessions repetitions (zero or more)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>Time interval for session activities (or &quot;0&quot; if start/end times are not specified)</td>
</tr>
</tbody>
</table>

#### Media description

<table>
<thead>
<tr>
<th>M</th>
<th>Media name and transport address</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>Media title</td>
</tr>
</tbody>
</table>

#### Media encoding

<table>
<thead>
<tr>
<th>Value</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>PCM, u-Law</td>
</tr>
<tr>
<td>3</td>
<td>GSM</td>
</tr>
<tr>
<td>4</td>
<td>G.723</td>
</tr>
<tr>
<td>5</td>
<td>DVI4</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Examples of SDP descriptions

Example 1
v = 0                      Version
o = mhandley 48949 48949 IN IP4 198.7.6.5 Creator and Session ID
s = Ok                     Name (Subject)
c = IN IP4 mediaserver.polito.it Connection information
b = CT:1000                 Max 1000Kbps (total)
m = audio 5004 RTP/AVP 0 3 Audio to port 5004, codec
                               PCMU or GSM

Esempio 2
v = 0                      Version
o = g.bell 87728 8772 IN IP4 132.151.1.19 Creator and Session ID
s = Come here, Watson!     Name (Subject)
u = http://www.ietf.org    E-mail
e = g.bell@bell-telephone.com
b = CT:64                   Bandwidth
b = CT:64                   Session start time
k = clear:manhole cover     Encryption: plaintext
m = audio 3456 RTP/AVP 96   Payload Type 96 => VDVI/8000/1
a = rtpmap:96 VDVI/8000/1
m = video 3458 RTP/AVP 31
m = application 32416 udp wb Application: whiteboard
a = orient:portrait         Whiteboard attribute
RTSP (Real-Time Streaming Protocol)

- Control of media streaming from a server
  - Video on Demand
  - Voice mailbox

- Commands to manage recording
  - DESCRIBE Request
  - SETUP Request
  - ISSUE media request
    - Play, record, Pause, etc
  - TEARDOWN Request
Main SIP components

IP network
- Gateway
- Location Server
- Redirect Server
- MCU
- Registrar Server
- Proxy Server
- Media Proxy

PSTN network

User terminals (user agent)
- UAS: User Agent Server
- UAC: User Agent Client
Main SIP components (2)

- **User Agent**
  - It must include
    - User Agent Client
    - User Agent Server

- **Registrar Server**
  - It keeps track of the mapping between user and its IP address (if any)

- **Proxy Server (outbound proxy)**
  - Used to have lighter user terminals
  - It is involved in outbound calls

- **Redirect Server**
  - It includes only an UAS
  - It re-directs incoming calls
  - Used to implement some "intelligent" policies
    - Office phone during working hours, e-mail otherwise, etc
Main SIP components (3)

- **Media Server**
  - Value added services
    - Voice mailbox
    - Entertaining music while waiting for a busy number

- **Media Proxy**
  - Used to “bounce” media streams when we cannot (or do not want to) send packets directly from one UA to the other

- **Location Server**
  - DNS, LDAP
  - address resolution

- **AAA server (Authentication, Authorization, Accounting)**
  - RADIUS server if often used

- **PSTN Gateway**
  - Translates signaling and media streams
Main SIP components (4)

- **Multipoint Control Unit (MCU)**
  - It is composed by two parts:
    - Multipoint Controller: in charge of session control (e.g. to decide which is the best codec to be used)
    - Multipoint Processor: in charge of data packet processing (mixing / stream switching, ...)
  - It is used in in the following cases:
    - Conference with 3 or more terminals with unicast transmission
    - Conference with 3 or more terminals with mixed transmission, both unicast and multicast
    - It makes no sense in case of pure multicast
Server consolidation

- The actual number of SIP server machine is normally low
  - Registrar Server, SIP Proxy e Redirect Server are often hosted in the same machine
  - In many cases, the Media Proxy is also hosted in the same machine
- Using existing servers
  - Location Server: included in the DNS server
  - AAA Server: included in the authentication corporate server
Distributed architecture (1)
Distributed architecture (2)

- Similar to e-mail
- The user physical location (IP address) does not impact the logical architecture

Advantages

- Scalability
- Easy management
  - Each domain can use its own rules (e.g. service available only to faculty members and not to the students)
- Privacy
- Sensible information (e.g. active/inactive state of a terminal) is not propagated outside the domain

Problem: how to define a mechanism to interconnect domains
Addressing

- The entity addressed is a user, not a terminal
- The Internet standard philosophy is adopted
  - username@domain.com
    - In many cases, it is identical to e-mail address
    - “domaino” may be in any possible call terminations
      - Domain server
      - Host name, IP address
  - telephone_no@gateway
    - In this case, the SIP header user=phone should be used to indicate that
      the destination is phone set
- Privacy
  - Implemented by means of filters in the user machine or in the server
  - The use of password in the URL is not a good choice, if the request is in
    plaintext
Addressing parameters

- Used to customize the call
  - Transport protocol
  - Multicast address
  - Time To Live

- Examples
  - sip:name.surname@domain.it;maddr=239.255.1.1;ttl=15
  - sip:name.surname@domain.it
  - sip:name.surname@domain.it;transport=tcp
  - sip:name.surname@domain.it?subject=project
  - sip:+1-212-555-1212:1234@gateway.com;user=phone
  - sip:alice@10.1.2.3
  - sip:alice@example.com
  - sip:alice%40example.com@gateway.com
  - sip:alice@registrar.com;method=REGISTER
DNS server configuration for SIP

- The address resolution uses the old DNS service
  - As for other services (e.g. e-mail), URI masks the actual destination address
  - A procedure is required to get the main parameters needed for a connection (IP address, port number, transport protocol, ...)

- Two DNS are involved
  - SRV
  - NAPTR
  - A/AAAA (used as usual in the final phase)
SRV record

- It defines the main parameters to access a given service
  - Server name, priority

- Syntax:

  
  _Service._Proto.Name TTL Class SRV Priority Weight Port Target

- Example:

  
  _sip._udp.foo.com 43200 IN SRV 10 10 5060 sipservr.foo.com.

  Higher Priority for small values
  Load balancing according this "weight"
Example of DNS registrations with redundant servers

foo.com IN SOA ns.foo.com. root.foo.com. (2003032001 10800 3600 604800 86400)

foo.com. 43200 IN NS ns.foo.com.

; ns.foo.com. 43200 IN A 10.0.0.20
server1.foo.com. 43200 IN A 10.0.0.21
server2.foo.com. 43200 IN A 10.0.0.22

;_sip._udp.foo.com. 43200 IN SRV 0 0 5060 server1.foo.com.
_sip._udp.foo.com. 43200 IN SRV 1 0 5060 server2.foo.com.
_sip._tcp.foo.com. 43200 IN SRV 0 4 5060 server1.foo.com.
_sip._tcp.foo.com. 43200 IN SRV 0 2 5060 server2.foo.com.
_sips._tcp.foo.com. 43200 IN SRV 0 0 5060 server1.foo.com.
_sips._tcp.foo.com. 43200 IN SRV 0 0 5060 server2.foo.com.

server2 is contacted only in the case server1 is unreachable

server1 is contacted a number of time that is as twice as the number of contacts for server2
NAPTR record

- It defines which transport protocol may be used
  - TCP, UDP, SCTP, TLS/TCP
- Not strictly required
- When present, there should be at least 3 records
  - For TLS/TCP, TCP, UDP (in this order)

Syntax:

domain-name TTL Class NAPTR order preference flags service regexp target

Example:

foo.com. 43200 IN NAPTR 60 50 "s" "SIP+D2U" "" _sip._udp.foo.com.

Higher priority for low values

load balancing proportional to “preference”
Example of DNS configuration with NAPTR

foo.com IN SOA ns.foo.com. root.foo.com. ( 2003032001 10800 3600 604800 86400 )

foo.com. 43200 IN NS ns.foo.com.

ns.foo.com. 43200 IN A 10.0.0.20
server1.foo.com. 43200 IN A 10.0.0.21
server2.foo.com. 43200 IN A 10.0.0.22

_sip._udp.foo.com. 43200 IN SRV 0 0 5060 server1.foo.com.
_sip._udp.foo.com. 43200 IN SRV 1 0 5060 server2.foo.com.
_sip._tcp.foo.com. 43200 IN SRV 0 4 5060 server1.foo.com.
_sip._tcp.foo.com. 43200 IN SRV 0 2 5060 server2.foo.com.
_sips._tcp.foo.com. 43200 IN SRV 0 0 5060 server1.foo.com.
_sips._tcp.foo.com. 43200 IN SRV 0 0 5060 server2.foo.com.

foo.com. IN NAPTR 0 0 "s" "SIPS+D2T" "" _sips._tcp.foo.com.
foo.com. IN NAPTR 1 0 "s" "SIP+D2T" "" _sip._tcp.foo.com.
foo.com. IN NAPTR 2 0 "s" "SIP+D2U" "" _sip._udp.foo.com.
Structure of SIP messages

It includes information about:
- users interested in this message,
- path that should be followed to reach the destination, etc.
Message types (1)

- **REGISTER**
  - Used to register a SIP user with a server
    - The destination server may be defined from the domain (e.g. *polito.it*)
    - It is possible that several UAs make this request, even at the same time
  - It can be sent using multicast (*all SIP servers, 224.0.1.75*)

- **INVITE**
  - Request to setup a call
    - Message sent to a server (proxy, redirect, terminal)
  - It includes an SDP description in the body
  - It may be sent also while a call is already in progress

- **ACK**
  - Positive notification of call established
  - Not the response message of the called user (in fact, it is sent by the caller)
  - It may include as SDP description of the session parameters; if not present, the parameters sent with the first INVITE are used
Message types (2)

- **BYE**
  - Signals the end of the call
  - It may be sent both during the call setup and when the call is in progress
  - The receiver of this message should stop immediately the output media streams

- **CANCEL**
  - Cancels a pending request for a call setup
  - It is used by servers that have sent duplicate requests
    - to communicate with an UAS that did not respond yet
    - to notify an UAC that one of the “fork” branches did not succeed (the BYE message received from the client is converted into CANCEL)
  - It may be generated by a client that decides to abort a call setup process
Message types (3)

- **OPTIONS**
  - Used to discover the UA capabilities
  - It is useful to understand the communication capabilities of a terminal

- **SUBSCRIBE**
  - Used to subscribe for the state of a different UA (e-presence)
  - It may be sent to the destination UA or to an e-presence server (according to the type of e-presence available in the domain)

- **NOTIFY**
  - Used to notify the UA state (e-presence)

- **MESSAGE**
  - Delivery of a message (text, XML, ...) between two UAs (instant messaging)
Main SIP headers (1)

- **From**
  - Indicates the SIP command initiator
  - It keeps the same value in the corresponding response message
    - From and To are not exchanged in the request and response messages

- **To**
  - Indicate the SIP request terminator

- **Contact**
  - Normally, it is used to carry the IP address to use for a direct UA-UA contact

- **VIA**
  - This field allows to keep track of the intermediate (SIP) system traversed
  - It is necessary if the call traverses several proxy servers.
  - The information in this header is added each time a new SIP system is traversed
Main SIP headers (2)

- **Record Routing**
  - It indicates that SIP messages should always go through the proxy (even when a connection has been established)
  - It is useful for NAT hole-punching

- **Call-ID**
  - Unique ID for
    - An invitation (in case of INVITE)
    - All the registration operations of the user (in case of REGISTER)

- **Cseq**
  - Command Sequence
  - Sequence number for each type of command (e.g., INVITE)
    - Not changed between request and response

- **Subject**
Main SIP headers (3)

- **Content-Type**
  - Payload type of the body of the SIP message (MIME type)
- **Content-Length**
  - Payload length in bytes
- **Content-Encoding**
  - Encoding applied to the payload contents (see MIME)
## Error codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1xx</td>
<td>Provisional</td>
<td>Request received, operation in progress</td>
</tr>
<tr>
<td></td>
<td>(Informational)</td>
<td></td>
</tr>
<tr>
<td>2xx</td>
<td>Success</td>
<td>Request received, understood and accepted</td>
</tr>
<tr>
<td>3xx</td>
<td>Redirection</td>
<td>Some additional operation is needed to complete the operation</td>
</tr>
<tr>
<td>4xx</td>
<td>Client Error</td>
<td>Syntax error or request not acceptable</td>
</tr>
<tr>
<td>5xx</td>
<td>Server Error</td>
<td>Correct syntax, but an internal server error occurred</td>
</tr>
<tr>
<td>6xx</td>
<td>Global Failure</td>
<td>No server can serve this request</td>
</tr>
</tbody>
</table>
### 1xx Provisional

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Trying</td>
</tr>
<tr>
<td>180</td>
<td>Ringing</td>
</tr>
<tr>
<td>181</td>
<td>Call Is Being Forwarded</td>
</tr>
<tr>
<td>182</td>
<td>Queued</td>
</tr>
</tbody>
</table>
2xx Success

- Only one code is defined
- The contents of the response body depends on the type of operation
  - BYE, CANCEL, INVITE, OPTIONS, REGISTER

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>OK</td>
</tr>
</tbody>
</table>
### 3xx Redirect

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>Multiple Choices</td>
</tr>
<tr>
<td>301</td>
<td>Moved Permanently</td>
</tr>
<tr>
<td>302</td>
<td>Moved Temporarily</td>
</tr>
<tr>
<td>303</td>
<td>See Other</td>
</tr>
<tr>
<td>305</td>
<td>Use Proxy</td>
</tr>
<tr>
<td>380</td>
<td>Alternative Service</td>
</tr>
</tbody>
</table>
# 4xx Client Error

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>Bad Request</td>
</tr>
<tr>
<td>401</td>
<td>Unauthorized</td>
</tr>
<tr>
<td>402</td>
<td>Payment Required</td>
</tr>
<tr>
<td>403</td>
<td>Forbidden</td>
</tr>
<tr>
<td>404</td>
<td>Not Found</td>
</tr>
<tr>
<td>405</td>
<td>Method Not Allowed</td>
</tr>
<tr>
<td>406</td>
<td>Not Acceptable</td>
</tr>
<tr>
<td>407</td>
<td>Proxy Authentication Required</td>
</tr>
<tr>
<td>408</td>
<td>Request Timeout</td>
</tr>
<tr>
<td>409</td>
<td>Conflict</td>
</tr>
<tr>
<td>410</td>
<td>Gone</td>
</tr>
<tr>
<td>411</td>
<td>Length Required</td>
</tr>
<tr>
<td>413</td>
<td>Request Entity Too Large</td>
</tr>
<tr>
<td>414</td>
<td>Request-URI Too Large</td>
</tr>
<tr>
<td>415</td>
<td>Unsupported Media Type</td>
</tr>
<tr>
<td>417</td>
<td>Temporary Request</td>
</tr>
<tr>
<td>420</td>
<td>Bad Extension</td>
</tr>
<tr>
<td>480</td>
<td>Temporarily not available</td>
</tr>
<tr>
<td>481</td>
<td>Call Leg/Transaction Does Not Exist</td>
</tr>
<tr>
<td>482</td>
<td>Loop Detected</td>
</tr>
<tr>
<td>483</td>
<td>Too Many Hops</td>
</tr>
<tr>
<td>484</td>
<td>Address Incomplete</td>
</tr>
<tr>
<td>485</td>
<td>Ambiguous</td>
</tr>
<tr>
<td>486</td>
<td>Busy Here</td>
</tr>
</tbody>
</table>
## 5xx Server Error

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>Internal Server Error</td>
</tr>
<tr>
<td>501</td>
<td>Not Implemented</td>
</tr>
<tr>
<td>502</td>
<td>Bad Gateway</td>
</tr>
<tr>
<td>503</td>
<td>Service Unavailable</td>
</tr>
<tr>
<td>504</td>
<td>Gateway Time-out</td>
</tr>
<tr>
<td>505</td>
<td>SIP Version not supported</td>
</tr>
</tbody>
</table>
## 6xx Global Failure

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>600</td>
<td>Busy Everywhere</td>
</tr>
<tr>
<td>603</td>
<td>Decline</td>
</tr>
<tr>
<td>604</td>
<td>Does not exist anywhere</td>
</tr>
<tr>
<td>606</td>
<td>Not Acceptable</td>
</tr>
</tbody>
</table>
Some examples

```
INVITE sip:called@dom.com SIP/2.0
Via: SIP/2.0/UDP gw.dom.edu
From: Fingal <sip:caller@dom.edu>
To: Patrik <sip:called@dom.com>
Call-ID: 1234567890@gw.dom.edu
Cseq: 1 INVITE
Subject: test
Content-Type: application/sdp
Content-Length: ...

v=0
o=caller 53655765 2353687637
IN IP4 123.4.5.6
s=Are you ready
c=IN IP4 host.dom.com
m=audio 5004 RTP/AVP 0 3 5
```

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP gw.dom.com
Via: SIP/2.0/UDP gw.dom.edu
From: Fingal <sip:caller@dom.edu>
To: Patrik <sip:called@dom.com>
Call-ID: 1234567890@gw.dom.edu
Cseq: 1 INVITE
Subject: test
Content-Type: application/sdp
Content-Length: ...

v=0
o=called 4858949 4858949
IN IP4 198.7.6.5
s=Yes
c=IN IP4 host.dom.edu
m=audio 5004 RTP/AVP 0 3
```
Transactions and dialogues

Transactions

- Request with associated response
- 2xx responses terminate the transaction
- The following SIP fields are not modified
  - From, To, Call-ID, Cseq

Dialogue

- Relation between UAs that begins when a “positive” response is received
- Messages may be exchanged directly (UA- UA), unless Record Route is selected
- The actual destination is included in contact:
Resolving a SIP address (1)

- It reduces to looking for a server SIP
  - The same procedure is used to search for either a user (INVITE message), or for a registration server (REGISTER message)

- Standard procedure:
  - Query NAPTR
  - Query SRV
  - Query A/AAAA
Resolving a SIP address (2)

- **Special cases**
  - The user provides the port number, or the numerical IP address
    - At most, the numerical IP address is found querying for A/AAAA records, with no NAPTR / SRV record
  - Non existent records
    - NAPTR: if it does not exist, SRV record is searched for
      - In many cases, experimental installations do not use it
      - Transport protocol selection is up to the user (often is UDP)
    - SRV: if it does not exist, A/AAAA record(s) is searched for
      - A standard port is used
Example of address resolution

- **Name to be resolved:** sip:bob@foo.com
- **NAPTR records found**
  
  ; order pref flags service regexp replacement
  IN NAPTR 50 50 "s" "SIPS+D2T" "" _sips._tcp.foo.com.
  IN NAPTR 90 50 "s" "SIP+D2T" "" _sip._tcp.foo.com
  IN NAPTR 100 50 "s" "SIP+D2U" "" _sip._udp.foo.com.

- **SRV records found**

  ;; Priority Weight Port Target
  IN SRV 0 1 5060 server1.foo.com
  IN SRV 0 2 5060 server2.foo.com

- **A/AAAA requests**
SIP registration (1)

User Agent

SIP Registrar

REGISTER sip:foo.com SIP/2.0

SIP/2.0 401 Unauthorized

WWW-Authenticate: Digest realm="foo.com",
nonce="418f9196437344c8ab31eb61604ad4fc2dd6d80"

Authorization: Digest username="bob@foo.com", realm="foo.com",
algorithm=md5, uri="sip:foo.com",
nonce="418f9196437344c8ab31eb61604ad4fc2dd6d80f",
response="92232ec9a0b4dc489a35599b1bd03e00e"
SIP registration (2)

**Procedure**
- A REGISTER message is sent, and the corresponding response is received.
- The first request fails, because the user should be authenticated.

**Authentication**
- It is required for the following reasons:
  - To avoid identity steal.
  - To make sure that only authorized users have access to some resources (e.g. VoIP - PSTN gateway).

**Registration data**
- Server: it is found with the usual procedure (NAPTR, SRV, ...).
- Username: it is divided in *username* and *realm*. 
SIP “trapezoid” (1)

1. INVITE
2. INVITE
3. 100 TRYING
4. INVITE
5. 100 TRYING
6. 180 RINGING
7. 180 RINGING
8. 180 RINGING
9. 200 OK
10. 200 OK
11. 200 OK
12. ACK
13. BYE
14. 200 OK

- Direct message sent to a destination stored in the “contact header” of the OK response.
- Messages are forwarded using the list recorded in VIA.

Media flow
SIP “trapezoid” (2)

- **Direct messages or through the proxy**
  - Initial messages are always sent through the proxies
    - Shipment of the messages is safer
    - Direct transmission could fail if the UA is behind a NAT
  - Later messages may or may not direct ones, according to whether or not Record Route has been selected in previous messages
  - Direct messages do not allow bandwidth control (not included in SIP)

- **Sending the first message from UA to its SIP proxy**
  - It could be eliminated if the UA implements the required procedure for server localization
  - Normally, the proxy is used to authenticate the user
The ENUM standard: E.164 addresses

- An infrastructure is needed to support the old E.164 addresses (phone numbers)
  - Traditional phone sets can only use numeric touch pads
  - Many IP phone sets still have only numeric touch pads
    - Users prefer to deal with "classical" phone sets
  - All these terminals should be able to connect with other users over Internet
ENUM standard: the problem (1)

Problem to be solved by ENUM:
- How it is possible for a VoIP station (SIP proxy or SIP Gateway) to determine dynamically:
  - Whether or not a given phone number is reachable over Internet?
  - If yes, what is its corresponding IP address?
  - If yes, which services are available with which preferences?

The PSTN network supports several service types (telephones, faxes, etc.)
- It is necessary that also the IP network would be able to find different addresses according to the type of service requested
- The terminal/gateway should use the suitable URI (according to the application and preferences)
ENUM standard: the problem (2)

- Scenarios
  - SIP call to different SIP user
  - Call from a VoIP gateway (from PSTN) to a SIP user
    - Example +39-011-5647008 → sip:+39-011-5647008@foo.com
    - A problem in both cases: how to associate a phone number and an IP address?
  - Problem not considered: SIP call to a PSTN user
    - Problems with call cost charging (for the last portion, beyond the VoIP/PSTN gateway)
- ENUM requires DNS
Working principles (1)

- It is necessary to deploy an ENUM application inside the terminal/gateway that does the following
  - Translation of an E.164 number into a domain name
  - Request to the DNS with the domain name

**E.164:** +39.011.5647008

\[
390110907027 \rightarrow 720709011093
\]

\[
7.2.0.7.0.9.0.1.1.0.9.3 \rightarrow \text{Domain: 7.2.0.7.0.9.0.1.1.0.9.3.e164.arpa}
\]
Working principles (2)

- DNS responds with a list of NAPTR records

```
$origin 8.0.0.7.4.6.5.1.1.0.9.3.e164.arpa.

; order pref flag service regexp for substitution
NAPTR 10 100 "u" "E2U+SIP" "^.*$!sip:info@foo.com!".
NAPTR 10 101 "u" "E2U+H323" "^.*$!h323:info@foo.com!".
NAPTR 10 102 "u" "E2U+MSG" "^.*$!mailto:info@foo.com!".
```

- List of standardized service names:
  - http://www.iana.org/assignments/enum-services

Selects all the characters (".*") included between the beginning ("^") and the end ("$") of the string
Scenarios for an ENUM call

- **PSTN – IP**
  - This exactly the case ENUM has been defined for
  - In this case, the PSTN–IP gateway translates the phone number in a SIP URI

- **IP – IP**
  - Originally not considered
  - A soft-phone (or a SIP proxy) translates the phone number into a SIP URI
  - Often soft-phones address auto-completion
    - +39-011-0907099 is forwarded to the proxy as +39-011-0907099@dominio.com
  - Hence, all the URIs beginning with “+” are translated, regardless of the domain (if any)
PSTN to IP connection

1. The user dials the E.164 number +39-011-0907027
2. The PSTN selects the appropriate gateway
3a. The gateway transforms into 7.2.0.7.0.9.0.1.1.0.9.3.e164.arpa
3b. DNS returns the list of associated SIP addresses
4a. The user@domain.com is selected, and DNS is queried for the SIP server of that domain
4b. DNS returns the IP address of the SIP server
5. The call is forwarded to the SIP server of domain.com
6. The call is forwarded to user@domain.com
IP to IP connection

1. The user dials the E.164 number +39-011-0907027

2. The application transforms the number into 7.2.0.7.0.9.0.1.1.0.9.3.e164.arpa and queries DNS

3. DNS returns all the records associated with the number

4. The application selects the suitable record (e.g. sip:user@domain.com) and it activates the standard SIP procedure to contact the destination
Creating a DNS infrastructure for ENUM

- Need for a world-wide infrastructure
  - Different visions about who should control it
    - Germany: politically and commercially neutral organization
    - China: against a TLD “.arpa”; domain assigned to ITU
    - France, Syria: domain assigned to ITU
  - Solution: domain e164.arpa
    - Under administrative control of ITU
    - Under operation control of IAB, which delegated RIPE NCC
  - Each nation holds a delegation for internal national organization
    - http://www.ripe.net/enum/request-archives/
  - Complex problems when several operators exist, with non disjoint numbering spaces
    - Number portability
VoIP with minimal ENUM presence (1)

- Managing ENUM is complicated
- For smaller networks, it is not necessary to use the whole ENUM standard
- It is possible to solve the problem of billing for calls to PSTN

If the number dialed is 4 digit long, append the domain `foo.com` to the destination, otherwise forward to SIP-PSTN gateway.
VoIP with minimal ENUM presence(2)

- Example of DNS configuration

```
$origin 0.9.0.1.1.0.9.3.e164.arpa.

; order pref flag service regexp for substitution
NAPTR 10 100 "u" "E2U+SIP" "!*map:*sip:\1@foo.com!".
```

This NAPTR record refers to the telephone domain +39-011-090xxxx

The substitution rule selects the input string, and integrates it with the requested URI
SIP and security

- User authentication
- Mechanisms to prevent or lessen Denial of Service
- Mechanisms to prevent Spam
- Encryption (optional) of the following parts
  - Message body
  - List of intermediate node traversed by the message
    - It may cause loops, because a node cannot check whether or not it has been already traversed by the same message
    - There are other mechanisms to prevent this
Conclusions

- Protocol that is quickly becoming popular
- Some problems
  - Too many commercial interests
    - Several pre-standard, incomplete, out of standard implementations
    - Problems in achieving agreements on hot problems
  - Problems to be solved (or still with experimental solutions)
    - Billing
    - IPv4/IPv6 interaction
    - NAT and firewall
    - Instant Messaging and E-presence (partially)